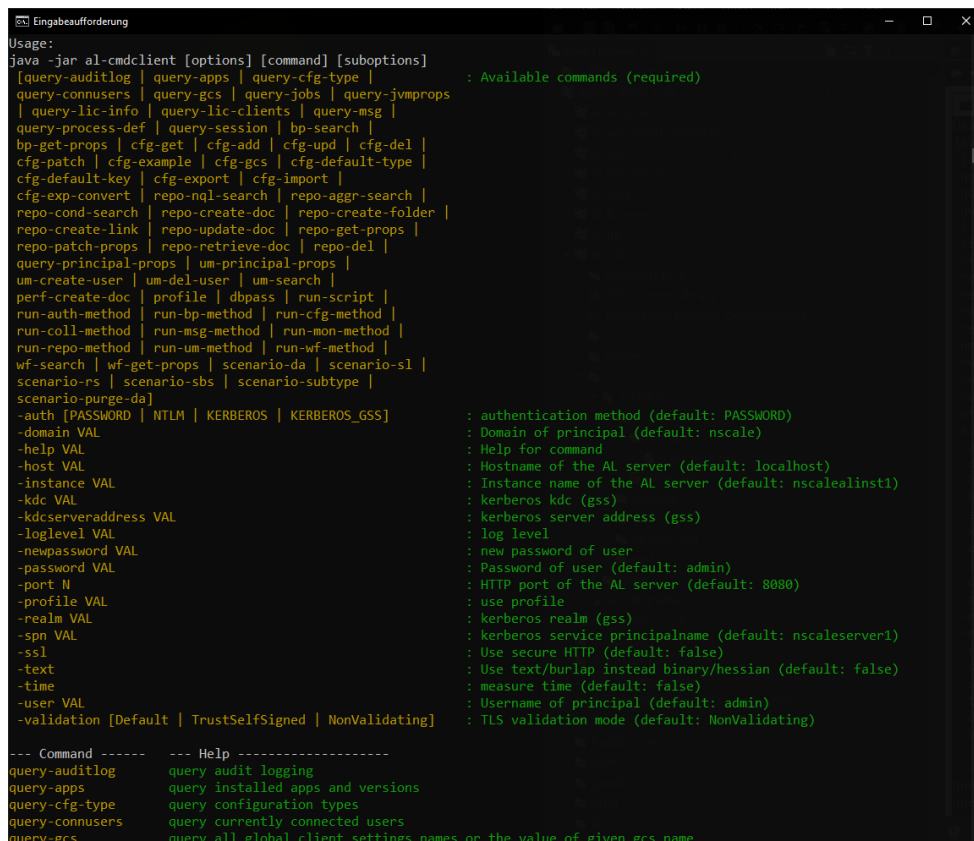


Kommandozeilen Tool al-cmdclient

Einleitung

Das Kommandozeilentool `al-cmdclient` wird im Application-Layer `bin` Verzeichnis mit ausgeliefert. Es kann mit der Batch-Datei `al-cmdclient.cmd` oder mit den Befehl `jre/bin/java -jar bin/al-cmdclient-9.0.xx.jar` gestartet werden.

Die JAR Datei kann auf einen anderen Rechner kopiert und benutzt werden. Das Programm benutzt intern den Advanced-Connector und kann daher auch Remote seine Kommandos ausführen.



```
Usage:
java -jar al-cmdclient [options] [command] [suboptions]
[query-auditlog | query-apps | query-cfg-type |      : Available commands (required)
query-connusers | query-gcs | query-jobs | query-jvmprops
| query-lic-info | query-lic-clients | query-msg |
query-process-def | query-session | bp-search |
bp-get-props | cfg-get | cfg-add | cfg-upd | cfg-del |
cfg-patch | cfg-example | cfg-gcs | cfg-default-type |
cfg-default-key | cfg-export | cfg-import |
cfg-exp-convert | repo-nql-search | repo-aggr-search |
repo-cond-search | repo-create-doc | repo-create-folder |
repo-create-link | repo-update-doc | repo-get-props |
repo-patch-props | repo-retrieve-doc | repo-del |
query-principal-props | um-principal-props |
um-create-user | um-del-user | um-search |
perf-create-doc | profile | dbpass | run-script |
run-auth-method | run-bp-method | run-cfg-method |
run-coll-method | run-msg-method | run-mon-method |
run-repo-method | run-um-method | run-wf-method |
wf-search | wf-get-props | scenario-da | scenario-sl |
scenario-rs | scenario-sbs | scenario-subtype |
scenario-purge-da]
-auth [PASSWORD | NTLN | KERBEROS | KERBEROS_GSS] : authentication method (default: PASSWORD)
-domain VAL : Domain of principal (default: nscale)
-help VAL : Help for command
-host VAL : Hostname of the AL server (default: localhost)
-instance VAL : Instance name of the AL server (default: nscalealinst1)
-kdc VAL : kerberos kdc (gss)
-kdcserveraddress VAL : kerberos server address (gss)
-loglevel VAL : log level
-newpassword VAL : new password of user
-password VAL : Password of user (default: admin)
-port N : HTTP port of the AL server (default: 8080)
-profile VAL : use profile
-realm VAL : kerberos realm (gss)
-spn VAL : kerberos service principalname (default: nscaleserver1)
-ssl : Use secure HTTP (default: false)
-text : Use text/burlap instead binary/hessian (default: false)
-time : measure time (default: false)
-user VAL : Username of principal (default: admin)
-validation [Default | TrustSelfSigned | NonValidating] : TLS validation mode (default: NonValidating)

--- Command ----- --- Help -----
query-auditlog      query audit logging
query-apps          query installed apps and versions
query-cfg-type      query configuration types
query-connusers     query currently connected users
query-gcs           query all global client settings names or the value of given gcs name
```

Figure 1. Programmaufruf

Aktuell sind die folgenden Punkte möglich

- Absetzen einer Query für bestimmte Objekte
 - AuditLog
 - Installiere Apps
 - Angemeldete Benutzer
 - Global-Client-Settings
 - Jobs
 - JVM Parameter
 - Messages
 - Sessions

- Virtual renditions
- Generisches Lesen und Ändern der Konfigurationen
 - cfg-get
 - cfg-add
 - cfg-upd
 - cfg-del
 - cfg-patch
 - cfg-default-type
 - cfg-default-key
- Import und Export von Konfigurationen
 - Import
 - Export
 - Konvertierung Binär ⇒ JSON
- Repository: Suchen und Anlegen von Objekte
 - NQL Search
 - Aggregated Search
 - String Condition Search
 - Dokument, Ordner und Link erzeugen
 - Properties patchen
- Scripting
 - JavaScript Datei ausführen, alternativ REPL (interaktiv)
 - Einzelne Server Service Methoden aufrufen
- Scenarios
 - Storage Layer und Rendition Server anlegen
 - Dokumentenbereich anlegen (mit SL und RS)
 - Dokumentenbereichs-Scenario ein- und ausschalten
 - SBS Scenario

Benutzung des Programms

Syntax

`al-cmdclient <global-options> subcommand <sub-options>`

Die globalen Optionen stellen die Verbindung zum Application Layer her

```
-auth [PASSWORD | NTLM | KERBEROS | KERBEROS_GSS] : authentication method
```

(default: PASSWORD)	
-domain VAL	: Domain of principal
(default: nscale)	
-help VAL	: Help for command [VAL]
-host VAL	: Hostname of the AL server
(default: localhost)	
-instance VAL	: Instance name of the AL
server (default: nscalealinst1)	
-kdc VAL	: kerberos kdc (gss)
-kdcserveraddress VAL	: kerberos server address
(gss)	
-loglevel VAL	: log level
-newpassword VAL	: new password of user
-password VAL	: Password of user
(default: admin)	
-port N	: HTTP port of the AL
server (default: 8080)	
-profile VAL	: use profile
-realm VAL	: kerberos realm (gss)
-spn VAL	: kerberos service
principalname (default: nscaleserver1)	
-ssl	: Use secure HTTP (default:
false)	
-time	: measure time (default:
false)	
-user VAL	: Username of principal
(default: admin)	

Kommandos:

- **Queries:** query-auditlog, query-apps, query-connusers, query-docarea, query-domain, query-gcs, query-jobs, query-jvmprops, query-ldap, query-lic-info, query-lic-clients, query-mds, query-msg, query-session, query-virtren, query-visirule
- **Konfiguration:** cfg-get, cfg-add, cfg-upd, cfg-del, cfg-patch, cfg-gcs
- **Import/Export:** cfg-export, cfg-import, cfg-exp-convert
- **Repository:** repo-nql-search, repo-aggr-search, repo-cond-search, repo-create-doc, repo-create-folder, repo-create-link, repo-get-props, repo-patch-props, repo-retrieve-doc, repo-del
- **Performance:** perf-create-doc
- **Scenarios:** scenario-da, scenario-sl, scenario-rs, scenario-sbs
- **Anmeldeprofil:** profile
- **Scripting:** run-script, run-auth-method, run-cfg-method, run-coll-method, run-msg-method, run-mon-method, run-um-method, run-wf-mothod

Hilfe

Ohne Argumente werden die möglichen Kommandos angezeigt. Die Option **-help** zeigt die Hilfe

eines Sub-Kommandos.

Hilfeaufrufe

```
# global help
al-cmdclient

# full help (all commands)
al-cmdclient -help

# help for one commands
al-cmdclient -help cfg-add
```

Anmeldung

Neben der Anmeldung mit Kennwort werden die SSO Verfahren NTLM und Kerberos unterstützt. Unter Windows kann die Anmeldesession mit `-auth KERBEROS` benutzt werden, unter Unix muss das platform-übergreifende Verfahren `KERBEROS_GSS` benutzt werden. Hier sind allerdings die Parameter `kdc`, `kdcserveraddress` und `spn` zu setzen. Die Verbindungsparameter lassen sich als Profile abspeichern.

Beispiele für Profile

Der `profile` Befehl kann Profile anlegen bzw. anzeigen. Die Basis-Option `-profile` kann ein Profil als Anmeldeparameter benutzen (statt explizit das Setzen von `host`, `port`, `user`, `password` oder `auth=TYPE`).

Anzeige der Profile:

```
al-cmdclient profile
#sample output: nscaleintern.json
```

Das gefundene Profil benutzen:

```
al-cmdclient -profile nscaleintern.json query-apps
```

Anzeige Profilinhalte:

```
al-cmdclient profile -name nscaleintern.json
{
  "host" : "nscaleintern.ceyoniq.com",
  "port" : 8080,
  "instance" : "nscalealinst1",
  "ssl" : false,
  "user" : "admin",
  "password" : "admin",
  "newpassword" : null,
  "domain" : "CT.com",
  "help" : null,
```

```

"auth" : "KERBEROS",
"servicePrincipalName" : "nscaleserver1",
"kerberosKDC" : "CT.COM",
"kerberosRealm" : "CT.COM",
"kdcServerAddress" : "001ctads2",
"logLevel" : null,
"profile" : null,
"command" : null
}

```

Queries

Anzeige der installierten Apps:

```

al-cmdclient query-apps

-- App-ID --
com.ceyoniq.nscale.businessapps.bl-app (APPS)
com.ceyoniq.nscale.businessapps.gov-app (APPS)
2.0.1201, 2.0.1201, 2.0.1201, 2.0.1201, 2.0.1300]

-- Versions --
[2.0.1201]
[2.0.1201, 2.0.1201, 2.0.1201, 2.0.1201, 2.0.1300]

```

Lesen von Global-Client-Settings, Anzeige durch Format-Parameter verbessern

```

# list all gcs
al-cmdclient query-gcs

# list one setting
al-cmdclient query-gcs -name Imaf_ALPolicies

# decode special format
al-cmdclient query-gcs -name AppMetaData -format JAVA_SER

```

Konfiguration

Lesen von PluginsSetting

```

# get plugin setting
al-cmdclient cfg-get -type PluginsSetting

```

Lesen und Ändern von LdapSetting

```

# print example LDAP setting
al-cmdclient cfg-example -type LdapSetting > ldap.setting

# import ldap setting
al-cmdclient cfg-add -type LdapSetting -file ldap.setting

```

```
# print ldap setting
al-cmdclient cfg-get -type LdapSetting -name CT

# patch property of ldap setting
al-cmdclient cfg-patch -type LdapSetting -name CT -set pageSize=1000

# patch complex property (use backslash oder two double quotes)
al-cmdclient cfg-patch -type LdapSetting -name CT -set "serverAddress={ \"host\" :
\"001ctads2\", \"port\" : 389 }"
```

Import und Export

Importieren einer Scenario-Manager Datei

```
# import nsb file
al-cmdclient cfg-import -file nscale-administrator-scenario.nsb -mode Alter

# convert binary file to JSON
al-cmdclient cfg-exp-convert -file nscale-administrator-scenario.nsb -out
scenario.json
```

Scenarios

Dokumentenbereich mit Rendition-Server und Storage-Layer anlegen. Dieser Befehl eignet sich gut für das Neuaufsetzen einer Datenbank bzw. NAPPL Instanz.

Aufruf:

```
# execute scenario "create document area"
al-cmdclient scenario-da -da DA
```

Repository

Es kann eine Suche mit NQL oder Fulltext Option abgesetzt werden. Eine String-Condition Suche kann pro Treffer-Seite den Bildschirminhalt stoppen (vergl. more Befehl). Als Ausgabeformat ist JSON oder Text möglich.

Suchen:

```
# NQL search
al-cmdclient repo-nql-search -da DA -nql "select displayname,resourcetype,contenttype
where identifier = 2 scope subtree"

# NQL aggregated search
al-cmdclient repo-aggr-search -da DA -nql "select count(identifier), fulltextstate
where fulltextstate > 0 order by count(identifier)"
```

```
# Search with string condition
al-cmdclient repo-cond-search -da DA -filter "identifier=4711"
-properties=displayname,resourceType,contenttype -subtree -pagesize=100
-sort=displayname -stop
```

Scripting

Es können einzelne Methoden oder ganze Skripte ausgeführt werden. Alternativ kann eine interaktive REPL Shell benutzt werden.

WARNING

Java 17: JavaScript Engine Nashorn fehlt

Mit der Java Version 17 fehlt die JavaScript Engine Nashorn, die noch in Java 11 LTS verfügbar war. Diese muss mit dem Classpath Argument hinzugefügt werden.

Scripting

```
# Simple method
al-cmdclient run-cfg-method -method getStandardDictionaryLocales

# Method parameter resourcekey (since 7.13)
al-cmdclient run-repo-method -method unlock "DA$NOTSET$204$1$NOTSET"

# Method parameter as json
al-cmdclient run-auth-method -method getAssignment "{ \"name\": \"default\",
\"areaName\": \"DA\" }"

# run javascript
al-cmdclient run-script -file myscript.js
```

REPL Shell

```
bash> al-cmdclient run-script

Oracle Nashorn [nashorn, Nashorn, js, JS, JavaScript, javascript, ECMAScript,
ecmascript]
The context variable 'session' can be used (class com.ceyoniq.nscale.al.core.Session).
Enter 'exit()' to terminate script.

> var cfgService = session.getConfigurationService()
> cfgService.getVersion()

7.11.9999.00000 (String)

> exit()
```